



(9)

## PROCESSOR FOR ELECTRONIC MUSICAL INSTRUMENT

Patent Number: JP2179697  
 Publication date: 1990-07-12  
 Inventor(s): SHIBA KOSUKE  
 Applicant(s):: CASIO COMPUT CO LTD  
 Requested Patent: JP2179697  
 Application Number: JP19880334164 19881229  
 Priority Number(s):  
 IPC Classification: G10H5/00  
 EC Classification:  
 Equivalents: JP7122796B

### Abstract

**PURPOSE:**To generate musical sounds according with functions of an electronic musical instrument and the degree of complication of processing by generating musical sounds in response to the interrupt signal generated with the sampling period from a timer interrupt control circuit and varying the sampling period.  
**CONSTITUTION:**A microcomputer performs interrupt handling to generate musical sounds at each time of generation of the interrupt signal from an interrupt control part 40 having a timer, a thereby keeping the musical sound sampling period during the operation. The generation interval of the interrupt signal generated by the control part 40, namely, the sampling period of musical sounds is variably programmed by the microcomputer 1. Thus, musical sounds are generated with the musical sound sampling period according with functions of the electronic musical instrument, the number of polyphonics, the manner of musical sound synthesis, and the degree of complication of processing.

Data supplied from the esp@cenet database - I2

4001 71101701  
 11 1111  
 (12/15/01)

(9)

⑩ 日本国特許庁(JP) ⑪ 特許出願公開  
⑫ 公開特許公報(A) 平2-179697

(2) 2月14日

⑬ Int. Cl.<sup>3</sup>  
G 10 H 5/00

識別記号 庁内整理番号  
6255-5D

⑭ 公開 平成2年(1990)7月12日

審査請求 未請求 請求項の数 2 (全14頁)

⑮ 発明の名称 電子楽器用処理装置

⑯ 特 願 昭63-334164

⑰ 出 願 昭63(1988)12月29日

⑱ 発 明 者 斯 波 康 祐 東京都西多摩郡羽村町栄町3丁目2番1号 カシオ計算機株式会社羽村技術センター内

⑲ 出 願 人 カシオ計算機株式会社 東京都新宿区西新宿2丁目6番1号

⑳ 代 理 人 井理士 町田 俊正

明 細 書

1. 発明の名称

電子楽器用処理装置

2. 特許請求の範囲

(1) 楽器を制御する入力処理するためのプログラムと演奏を生成するためのプログラムを記憶するプログラム記憶手段と、

前記プログラム記憶手段のアドレスを制御するアドレス制御回路手段と、

演奏の生成に必要なデータを記憶するデータ記憶手段と、

演算回路手段と、

前記プログラム記憶手段のプログラムの各命令を解釈して前記各手段の動作を制御するオペレーション制御回路手段と、

を備えるマイクロコンピュータから成り、更に、前記マイクロコンピュータは演奏のサンプリング周期でインタラプト信号を発生するタイマー

インタラプト制御回路手段を有し、このタイマーインタラプト制御回路手段からのインタラプト信号に応じて前記演奏を生成するためのプログラムが前記マイクロコンピュータにおいて実行されることにより演奏が生成され、更に前記タイマーインタラプト制御回路手段において前記インタラプト信号が発生する間隔である前記演奏のサンプリング周期を可変にプログラム可能にしたことを特徴とする電子楽器用処理装置。

(2) 請求項1記載の電子楽器用処理装置において、前記マイクロコンピュータは集積回路チップで構成され、このチップ上にデジタルの演奏信号をアナログ信号に変換するデジタル・アナログ変換器と楽器を制御する入力を受けるポートが更に設けられることを特徴とする電子楽器用処理装置。

3. 発明の詳細な説明

【発明の技術分野】

この発明は電子楽器用処理装置に関し、特に電

子楽器用処理装置における楽音合成の構造的なアーキテクチャに関する。

#### 【発明の背景】

近年、電子楽器はコンピュータ化されている。しかし、大量で高速のデータ演算が必要な楽音の生成に係る部分は音源回路と呼ばれる専用構造のハードウェアで行われており、マイクロコンピュータは楽器への制御入力（鍵盤やコンソールパネルからの入力、MIDIその他の外部制御入力、内部または外部の演奏メモリからの入力等）を処理し、音源回路に適したコマンドを音源回路に転送するに留まっている。

楽音生成処理は音源回路ハードウェアで行い、楽器の制御入力の処理をマイクロコンピュータで行うという電子楽器のシステムアーキテクチャにはいくつかの問題がある。第1に音源回路ハードウェアは楽音パラメータを処理する種々の処理段階の随所にデータを一時的に保持する記憶装置、演算を行う演算回路から構成されるので、必

然的に回路規模が大きくなる。代案的にはマイクロコンピュータの2倍程度の規模となる。第2に音源回路ハードウェアで行う楽音合成の仕方をマイクロコンピュータで可変に制御するに限界がある。例えば、同時発音数（ポリフォニック数）はハード的に固定されているのでマイクロコンピュータからのコマンドで変えることはできない。また、楽音のサンプリング周波数も音源回路のハードウェアにより固定されているため、マイクロコンピュータからのコマンドで変更することはできない。このような制御の限界は新しい音源回路ハードウェアを設計する場合の障壁になる。すなわち、しばしば大規模な回路変更が余儀なくなり、多大の開発時間、労力を要する。更に、マイクロコンピュータと音源回路ハードウェアとの通信プロトコルあるいはインタフェース（転送方式、コマンドのセット等）についても、音源回路ハードウェアごとに検討し直し、開発し直す必要がある。

以上の理由から、本件出願人は音源回路ハード

ウェアを使用することなく、マイクロコンピュータのプログラム制御だけで楽音を生成することのできる新しい電子楽器のシステムアーキテクチャの研究を続け、ついに、その実現を見た。

新しい電子楽器のシステムアーキテクチャを実現する上で解決すべき課題として、処理すべき仕事量がそのときそのときのシステムの状態によって変動せざるを得ないマイクロコンピュータにおいて、楽音のサンプリング周波数、すなわちマイクロコンピュータがデジタル楽音信号を生成する間隔をどのようにしたら有効に維持することができるか、という問題があった。更に、電子楽器の種々の使用、例えば楽音の合成の仕方、その処理の複雑さの度合、ポリフォニック数などに合わせて最適な楽音サンプリング周波数を容易に設定できるような機構が望まれた。

#### 【発明の目的】

したがって、この発明の目的は従来のような音源回路ハードウェアを必要とすることなくプログ

ラム制御により、マイクロコンピュータ自身が楽音を生成することができ、かつプログラム制御に起因する問題、特に動作時の楽音サンプリング周期の確立の問題を解決するとともに電子楽器の機能に関する種々の仕様変更、設計変更、ポリフォニック数の変更、楽音合成方式の変更、楽器制御処理の変更等に対してただちに対応できるような最適な楽音サンプリング周期の選定機能を実現した電子楽器用処理装置を提供することである。

#### 【発明の構成、作用】

この発明によれば、上記の目的を達成するため、楽器を制御する入力を処理するためのプログラムと楽音を生成するためのプログラムを記憶するプログラム記憶手段と、このプログラム記憶手段のアドレスを制御するアドレス制御回路手段と、楽音の生成に必要なデータを記憶するデータ記憶手段と、演算回路手段と、前記プログラム記憶手段のプログラムの各命令を解説して前記各手段の動作を制御するオペレーション制御回路手段

とを調えるマイクロコンピュータから成り、更に、前記マイクロコンピュータは演奏のサンプリング周期でインタラプト信号を発生するタイマーインタラプト制御回路手段を有し、このタイマーインタラプト制御回路手段からのインタラプト信号にตอบสนองして前記演奏を生成するためのプログラムが前記マイクロコンピュータにおいて実行されることにより演奏が生成され、更に前記タイマーインタラプト制御回路手段において前記インタラプト信号が発生する間隔である前記演奏のサンプリング周期を可変にプログラム可能にしたことを特徴とする電子楽器用処理装置が提供される。

すなわち、この発明では、タイマーインタラプト制御回路手段からインタラプト信号によりマイクロコンピュータが行っている仕事を中断させ、演奏を生成するプログラムを実行させているので、演奏のサンプリング周期ごとに演奏のサンプルを生成するという課題がプログラム制御方式において有効に解決される。すなわち、演奏を生成する処理は他の処理に優先され、タイマーインテ

ラプト信号ごとにカウント値をデクリメントする。カウント値がゼロになるとカウンタからインタラプト信号が発生し、マイクロコンピュータに割込を要求する。更にこのインタラプト信号によりプリセットデータ（保持されている）が再度、カウンタにセットされる。以上により、マイクロコンピュータが望むサンプリング周期で、タイマーインタラプト制御回路手段から割込がかかり、その都度マイクロコンピュータは演奏生成処理を実行する。

演奏生成処理（インタラプト処理）はタイマーインタラプト制御回路手段から次のインタラプト信号が出力されるより前に、完了しなければならない。これができなければ、演奏のサンプリング周期を維持できない。演奏生成処理に要する時間は、その内容、例えばポリフォニック数、演奏合成の仕方、処理の複雑さの成合に依存する。換言すれば、演奏生成処理プログラムの実行時間に依存する。例えば、他の条件を同じとし、ポリフォニック数を半分にすれば、その演奏生成処理プロ

ラプト制御回路手段からインタラプト信号が発生する都度、実行される。タイマーインタラプト制御回路手段は基本的にタイマー（カウンタ）で構成されるわけであるから、そのインタラプト信号の発生間隔は非常に安定している。またインタラプト信号の発生ごとに演奏のサンプルが生成されるわけであるから、インタラプト信号の発生間隔は演奏のサンプリング周期にほぼならない。更に、この発明によれば、このタイマーインタラプト制御回路手段はプログラム可能である。したがって、ポリフォニック数、演奏の合成の仕方などに合わせて、最適の間隔でタイマーインタラプト制御回路手段からインタラプト信号を発生させることができる。例えば、タイマーインタラプト制御回路手段をプリセット可能なダウンカウンタで構成したとする。マイクロコンピュータは適当なとき（例えばパワーオン時の初期設定処理時）に、このカウンタにサンプリング周期を表すデータをプリセットする。カウンタはクロック（水晶発振器等のマスタークロック源からの信号

プログラムの実行時間はほぼ半分にになる。この場合、演奏のサンプリング速度を速くしても問題は無い。むしろ、デジタル・アナログ変換器の交換を考慮すれば、サンプリング速度を速くした方がよい。一方、用途によっては、高品質でスペクトルの豊かな演奏が要求される場合もある。この場合、ポリフォニック数は犠牲になるが、1つの音を複雑な処理で合成して要求される条件にかなう演奏を提供する必要がある。このような場合にも、それに適した演奏サンプリング周期が考えられる。この発明は、以上のような様々な使用条件を考慮して、それぞれの条件に合った演奏サンプリング周期での演奏の生成を可能にしたものである。

一構成例において、上記マイクロコンピュータは集積回路チップで実現され、このチップ上に上記手段に加え、生成したデジタル演奏信号をアナログ信号に変換するデジタル・アナログ(D/A)変換器と演奏を制御する入力を受けるポートも実装される。したがって、音源回路ハードウェア

アが不要であるので、全体の回路規模を大幅に縮小することができる。従来の場合、音源回路LSIチップの回路規模が大きいので、チップ製造における歩留りに限界がある(歩留りはチップ面積にほぼ反比例する)。したがって、この発明は電子楽器の製造コストを大幅に下げることができる。

#### 【実施例】

以下、図面を参照してこの発明の実施例を説明する。

本実施例に係る電子楽器の全体構成を第1図に示す。装置全体の制御はマイクロコンピュータ1により行われる。特に、この発明に従い、楽器の制御入力処理のみならず、楽音を生成する処理もマイクロコンピュータ1で実行され、楽音生成用の音源回路ハードウェアは必要としない。鍵盤2と機能キー3とから成るスイッチ部4は楽器の制御入力部であり、スイッチ部4から入力された情報はマイクロコンピュータ1で処理される。マ

イクロコンピュータ1の生成したアナログ変換後の楽音信号はローパスフィルタ5でフィルタリングされ、アンプ6で増幅され、スピーカ7を介して放音される。電源回路8はマイクロコンピュータ1、ローパスフィルタ5、アンプ6に必要な電圧を供給する。

上記マイクロコンピュータ1の内部構成を第2図にブロック図で示す。図示の各要素はワンチップ上に実装されている。実際に製作したものは5×5mmのチップサイズで、8音ポリフォニックの同時発音数をもち、楽音合成方式はPCM(波形成み出し方式)であるが、本発明は他のポリフォニック数、他の楽音合成方式にも適用できる。

制御用ROM31には楽器の各種制御入力処理するプログラムと楽音を生成するプログラムが記憶されており、ROMアドレス制御部39からROMアドレスデコード32を介して指定されたアドレスのプログラム語(命令)を順次出力していく。なお、具体的実施例では、プログラム語長

は28ビットであり、プログラム語の一部が次に読み出されるべきアドレスの下位部(ページ内アドレス)としてROMアドレス制御部39に入力されるネクストアドレス方式となっているが、プログラムカウンタ方式のものにも本発明を適用し得る。RAMアドレス制御部33は制御用ROM31からの命令のオペランドがレジスタを指定している場合に、RAM34内の対応するレジスタのアドレスを指定する。RAM34はレジスタ群であり、汎用演算、フラグ演算、楽音の演算等に使用される。加減算器及び論理演算部35と乗算器36は制御用ROM32からの命令が演算命令のときに用いられる。特に乗算器36は楽音波形の演算に使用しており、そのための最適化として第1と第2のデータ入力(例えば16ビットデータ)を乗算して入力と同じ長さ(16ビット)のデータを出力するようになっている。上記RAM34、加減算器35、乗算器36により、演算回路(AU)が構成される。制御データ兼波形用ROM37にはピッチデータ、エンベロープデー

タ(レート、レベル)などの各種楽音制御パラメータと、PCM(パルス符号変調)の楽音波形データが記憶されている。エンベロープデータと楽音波形データは楽音の音色ごとに用意される。オペレーション解析部(オペレーション制御回路)38は制御用ROM31からの命令のオペコードを解説し、指示されるオペレーションを実行するために、回路の各部に制御信号を送る。

所定時間ごとに制御用ROM31の楽音生成プログラムを実行するため、本装置ではタイマー・インタラプトを採用している。すなわち、タイマーを有するインタラプト制御部40により、一定時間ごとにROMアドレス制御部39に制御信号(インタラプト信号)を送り、この信号により、ROMアドレス制御部39は次に行うメインプログラムの命令のアドレスを遊離(保持)し、楽音の生成が行われるインタラプト処理プログラム(サブルーチン)の先頭アドレスを代りにセットする。これにより、インタラプト処理プログラムが開始される。インタラプト処理プログラムの最

後にはリターン命令があるので、このリターン命令がオペレーション解析部38で解説された時点で、ROMアドレス制御部39は退避してあったアドレスを再度セットし、メインプログラムに復帰する。

上記インタラプト制御部40はこの発明に従いプログラマブルである。すなわち、インタラプト制御部40のインタラプト信号の発生間隔(演奏サンプリング周期)をマイクロコンピュータ1で自由に設定できる。詳細は後述する。

入力ポート41と出力ポート42は鍵盤2、機能キー3のキースキャンのために使用される。インタラプト処理プログラムにおいて生成された演奏はデジタル/アナログ変換部43でアナログ信号に変換され、外部に出力される。

第3図(A)に本実施例のマイクロコンピュータ1のメインプログラムのフローを示す。A1は電源投入時のイニシャル処理であり、マイクロコンピュータ1のRAM(レジスタ群)34のクリアや、リズムテンポ等の初期値の設定等を行う。

演奏データ兼波形用ROM37からリズムデータを順次読み出し、発音処理A9のためのキーアサイン処理を行う。フロー間タイマー処理A8では、メインフローで必要なイベントのタイミングを知るために、フロー間時間(これは、フローを一回する間に実行されたタイマーインタラプトの回数を計数することで得られる。この計数処理は後述のインタラプトタイマー処理B3で行われる。)を基に演算を行い、エンベロープ用タイマー(エンベロープの演算周期)やリズム用の基準値を得る。発音処理A9ではA5、A6、A7でセットされたデータから、実際に演奏を発音させるための各種演算を行い、結果をRAM34内の音源処理レジスタ(第8図)にセットする。A10は次のメインフローのバスのための準備処理であり、今回のバスで得た押鍵状態への変化を示すNEW ON状態をON中にしたり、離鍵状態への変化を示すNEW OFF状態をOFF中に変える等の処理を行う。

演奏の生成が行われるインタラプト処理プログ

更に、このイニシャル処理において、インタラプト制御部40に対する演奏サンプリング周期データの設定が行われる。A2でマイクロコンピュータ1は出力ポート42からキー走査のための信号を出力し、スイッチ部4の状態を入力ポート41から取り込むことにより、機能キー、離鍵キーの状態をRAM34のキーバッファエリアに記憶する。A3ではA2で得た機能キー3の状態から、状態の変化した機能キーを識別し、指示される機能の実行を行う(例えば、演奏番号のセット、エンベロープ番号のセット、リズム番号のセット等)。A4ではA2で得た鍵盤2の状態から、変化した鍵(押鍵、離鍵)を識別する。次のA5でA4の処理結果から、発音処理A9のためのキーアサイン処理を行う。A6では機能キー3でデモ演奏キーが押鍵されたとき制御データ兼波形用ROM37から、デモ演奏データ(シーケンサデータ)を順次読み出し、処理することにより、発音処理A9のためのキーアサイン処理等を行う。A7ではリズムスタートキーが押鍵されたとき制

ラムのフローを第3B図に示す。B1で前回のインタラプトの音源処理B2で生成してある演奏波形データ(8音分の累算波形値)をD/A変換部43に送出する。これにより、一定周期でD/A変換部43に演奏のサンプルが与えられることになる。次の音源処理B2で各チャンネルの演奏が生成され累算される。従来はこの処理を音源回路ハードウェア上で行っていた。詳細は後述する。次のインタラプトタイマー処理B3ではインタラプトが所定時間ごとにかかることを利用して、フロー間計時用のタイマーレジスタ(RAM34内)を満満の程度、プラス1する。

なお、この実施例ではインタラプト処理プログラム内ではメインプログラムで書込を行うレジスタについては、内容の書替を行わないようにしている。通常のインタラプト処理の開始時と終了時に行われるレジスタの退避と回復の処理は不要である。すなわち、RAM34上のレジスタは演奏処理に関係するレジスタとその他の処理に関係するレジスタとが独立している。メイン

ログラムからインタラプト処理への移行が可及的に遅れなしで行われる。

音源処理B2の詳細を第3C図に示す。C1で波形加算用RAM領域(第6図参照)をクリアした後、8チャンネル分の処理C2~C9を順番に行っている。各チャンネル処理の最後で、チャンネルの乗音波形値が波形加算用RAM領域のデータに加算される。

第4図は、時間によって実施例の動作の流れを編いたものである。A、B、C、D、E、Fはメインプログラム(第3A図)の断片であり、一定時間(乗音サンプリングタイム)ごとにインタラプト処理(第3B図)が実行される。タイムチャートで示すと第5図のようになる。図示のように、インタラプトに入る都度、D/A変換部43に乗音波形番号が入力され、対応するアナログ信号が外部に出力されていく。

第3C図のC2~C9の処理を1チャンネル分について詳細に示したのが第7図である。チャンネル処理は大きく分けてエンベロープ処理(D

1~D7)と波形処理(D8~D21)から成る。

第8図はエンベロープ処理で生成されるエンベロープを示したものである。1つの乗音のエンベロープはいくつかのスタップ(セグメント)から成っている。図では4セグメントで示してある。図中の $\Delta x$ はエンベロープのサンプリング周期であり、 $\Delta y$ はエンベロープ値の変化幅である。チャンネルのエンベロープ処理(D1~D7)では、サンプリングタイムごとのエンベロープの計算とスタップの目標レベルに適したかどうかのチェックを行っている。一致したときには現在エンベロープレジスタ(第6図参照)に目標値が設定されるので、メインプログラムの乗音処理A9内でそれを検知して、次のスタップのエンベロープのためのデータ( $\Delta x$ 、 $\Delta y$ 、目標エンベロープ値)を各レジスタにセットしている。

詳細に述べると、D1でエンベロープの演算周期 $\Delta x$ と比較するためのタイマーレジスタをインタラプトごとにインクリメントし、D2で $\Delta x$ と

一致したときD3でエンベロープ変位分のデータ $\Delta y$ の加減算フラグ(符号ビット)をテストしてエンベロープが上昇中か下降中かを判別し、D4、D5でそれぞれ現在エンベロープの減算または加算を行う。D6で現在エンベロープが目標エンベロープ値に適したかどうかをチェックし、適しておれば、現在エンベロープに目標レベルをセットする。これによりメインプログラムの乗音処理A9で次のエンベロープスタップのデータがセットされることになる。また乗音処理A9でゼロの現在エンベロープを読んだときには乗音の終了として処理される。

次に、波形処理D8~D21について述べる。波形処理では、現在アドレスの整数部を使って波形ROMから読み合う2つアドレスの波形データを読み出し、(整数部+小数部)で示される現在アドレスに対して想定される波形値を補間求めている。補間が必要な理由は、インタラプトによる波形サンプリング周期が一定であり、アドレスの加算値(ピッチデータ)が乗音への応用上、あ

る音域にわたるためである(音階音しか出力しない乗音で音階音ごとに波形データを用意すれば補間の必要はないが許容できない記憶容量の増大となる)。補間による音色の劣化、歪みは高音域の方が著しいため、通常は、原音の記録サンプリング周期より高速の周期で原音を再生する。この実施例では原音(A4)再生の周期を2倍にしている(第9図)。したがって、アドレス加算値が0.5のとき、A4の音が得られるようになっている。この場合、A#4ではアドレス加算値は0.529となり、A3のとき、1となる。これらのアドレス加算値はピッチデータとして調律データ演算ROM37に記憶されており、押鍵時には乗音処理A9において、鍵に対応するピッチデータと選択されている音色の波形スタートアドレス、波形エンドアドレス及び波形ループアドレスがRAM34の対応するレジスタ、すなわち、アドレス加算値レジスタ、スタートアドレス兼現在アドレスレジスタ、エンドアドレスレジスタ、ループアドレスレジスタにセットされる。

参考までに、第10図に時間に対する補間波形データを示す。図中、白丸は波形ROMのアドレスにある波形データ値、黒丸は補間値を示している。

補間の方式はいろいろあるが、ここでは直線補間を採用している。第7図の波形生成処理D8～D21を詳細に述べると、まず、D8で現在アドレスにアドレス加算値を加算して新しい現在アドレスを得る。D9で現在アドレスとエンドアドレスを比較し、現在アドレス>エンドアドレスならば、D10、D11により、現在アドレス<エンドアドレスのときはD12により、物理上（番地上）または論理上（動作上）の次のアドレスを計算し、D14でその整数部により波形ROMをアクセスして次回波形データを得る。ループアドレスは動作上エンドアドレスの次のアドレスである。すなわち、第9図の場合、図示の波形は繰り返して読み出される。したがって、現在アドレス=エンドアドレスのときは次のアドレスとしてループアドレスの波形データを読み出す（D13）。

D15、D16により、現在アドレスの整数部で波形ROMをアクセスして今回の波形データを読み出す。次に、D17で次回波形値から今回波形値を減算し、D18でその差に現在アドレスの小数部を乗算し、その結果をD19で今回の波形値に加えることにより、波形の直線補間値を求める。この直線補間したデータに現在エンベロープ値を乗算してチャンネルの乗音データ値を得（D20）、それを波形加算用レジスタの内容に加えて乗音データを累算する（D21）。

ここまでの、明らかなように、マイクロコンピュータ1は、インタラプト制御部40からのインタラプト信号の発生の都度、インタラプト処理を行い、乗音を生成することにより、動作中の乗音サンプリング周期を維持している。更に、この発明に従い、インタラプト制御部40の発生するインタラプト信号の発生間隔、すなわち、乗音のサンプリング周期はマイクロコンピュータ1により可変にプログラムすることができる。プログラム可能なインタラプト制御部40の構成は種々考

えられる。以下、2つの例について説明する。

第11図に第1構成例のインタラプト制御部40を示す。このインタラプト制御部は2通りの乗音サンプリング周期でインタラプト信号INTFを発生することができる。詳細に述べると、インタラプト制御部には第13図に示すようなクロック信号φが入力される。このクロック信号φはクロック発生部の直接の出力あるいは、その出力を分周回路に通した後の出力である。図示のようにクロック信号φはリプルカウンタを構成する8つのバイナリカウンタ11-2～11-7の最下位のバイナリカウンタに入力される。第1のORゲート2-8にはすべてのバイナリカウンタ11-2～11-7の出力が入力され、第2のORゲート2-9には最上位ビットのバイナリカウンタ11-7を除いたバイナリカウンタ11-2～11-6の出力が入力される。したがって、第1のORゲート2-8の出力は128図の基本クロックφに対し、1回“0”となり、第2のORゲート2-9の出力は64図の基本クロックφに

対し、1回“0”となる。この第1と第2のORゲート2-8、2-9のいずれを選択するかをプログラム制御で設定できるようにすればよい。そのために、ラッチ11-1がある。このラッチ11-1は上述した実施例の場合、初期設定処理A-1のなかで“0”または“1”に設定される。すなわち、データバスの所定のビット線に128/φ（φは基本クロックφの周波数）の乗音サンプリング周期を指示するデータ“0”、またはその半分の長さの乗音サンプリング周期を指示する“1”が受った状態でコマンド解析部38からの制御信号がラッチ11-1のクロック入力に加って、データがラッチされる。ラッチ11-1の出力は第1のORゲート2-8の出力が入力されるORゲート11-10に入力され、またラッチ11-1の出力をインバータ11-11で反転した信号が、第2のORゲート2-9の出力が入力されるORゲート11-12に入力される。ORゲート11-10とORゲート11-12の出力はNANDゲートに入力され、その出力がイ



インタラプト番号INTFとなる。したがって、ラッチ11-1にビット“0”を設定したときには、第1のORゲート11-8の出力が選択され、基本クロックφの128個ごとに1回、インタラプト番号INTFが“1”アクティブになり、ラッチ11-1にビット“1”を設定したときには、第2のORゲート11-9の出力が選択され、インタラプト番号INTFは基本クロックφの84個ごとに1回“1”アクティブとなる。

第12図に第2構成例のインタラプト制御部40を示す。このインタラプト制御部は第11図の構成より細かく高音サンプリング周期を調整することができる。詳細に述べると、1ビットラッチ11-1の代りに、データバスの4ビットに接続される4ビットのラッチ12-1が使用される。したがって、マイクロコンピュータ1はこのラッチ12-1に0~15の任意の値が設定可能である。この4ビットラッチ12-1の出力はコンパレータ12-8のB入力に入力される。コン

パレータ12-8のA入力には第11図のリブルカウンタと同様な8ビットのリブルカウンタ12-2~12-7の出力のうち、上位4ビットのバイナリカウンタ12-4~12-7の出力が入力される。コンパレータ12-8は、A入力とB入力が等しいとき“1”となり、インタラプト番号INTFを発生する。このインタラプト番号INTFはリブルカウンタ12-2~12-7の各リセット入力にも入力されており、インタラプト番号INTFの発生時にリブルカウンタ12-2~12-7をオールゼロにリセットする。したがって、第12図の構成の場合、インタラプト番号INTFの発生周期(高音のサンプリング周期)を任意の $8\alpha/\phi$ ( $\alpha=1\sim$ )に設定することができる。例えば、4ビットラッチ12-1が8のとき(B入力=1000)は、リブルカウンタ12-2~12-7の値が1000000(2進)になるごとにインタラプト番号INTFが発生する。すなわち、84個の基本クロックφごとに1度の頻りとなる。4ビットラッチ12-1が

9のときなら、72個のクロックφごとに1度の割合となる。

以上で実施例の説明を終えるが、この発明の適用を逸脱することなく種々の変形、変更が可能である。例えば、ある用途では、電子楽器に複数の異なる高音合成能力をもたせることが考えられる。この場合、それぞれの高音合成に対するプログラムがプログラムメモリ(制御用ROM31)に用意される。高音合成方式ごとに最適なサンプリング周波数があるとする。またモード切換スイッチにより、どの高音合成方式を使用するかが、ユーザーで選択できるとする。そのために、メインフロー上でモード切換スイッチの検査を行い、選択されている高音合成番号をメモリ(RAM34)にセットする。更に、高音生成処理ごとのサンプリング周波数データのテーブル(例えば制御用ROM41に置かれる)をアクセスし、選択された高音合成番号に対応するデータを見つけ、インタラプト制御部40にセットする。インタラプト制御部40からインタラプト番号が発生

したら、夫々の高音合成処理ルーチンのエントリポイントのテーブル(例えば、制御用ROM41に置かれる)をアクセスし、選択された高音合成番号に対応するエントリポイントを見つけ、その位置から開始する高音生成プログラムを実行する。

その他、種々の変更が可能である。

#### 【発明の効果】

以上のように、この発明ではプログラム制御によりマイクロコンピュータ自身が高音を生成する場合において、高音合成処理をタイマーインタラプト制御回路手段からのインタラプト番号の合図で開始させるとともに、インタラプト制御手段のインタラプト番号の発生周期をマイクロコンピュータからプログラムできるようにしているので、動作時に高音のサンプリング周期を維持することができる一方で、電子楽器の機能、ポリフォニック数、高音合成の仕方、処理の複雑さの度合に合った高音のサンプリング周期で高音を生成する

ことができる。更に、この実現のために、格別の回路の変更は不要であり、自由度の高い電子楽器の設計、開発が可能となるという利点がある。

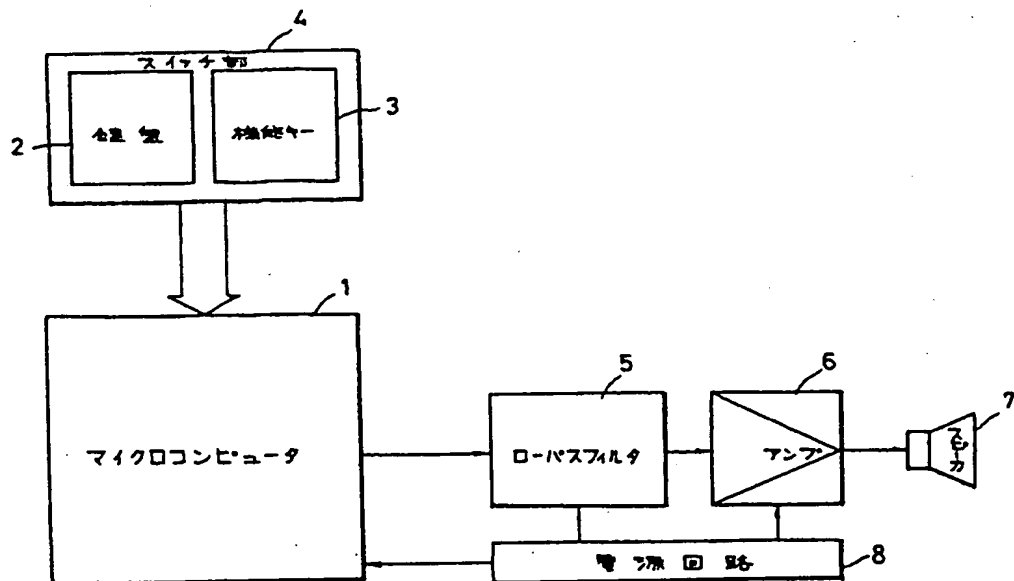
#### 4. 図面の簡単な説明

第1図はこの発明の実施例に係る電子楽器の全体構成図、第2図は実施例のマイクロコンピュータの構成を示すブロック図、第3A図はマイクロコンピュータのメインプログラムのフローを示す図、第3B図は楽音の生成が行われるインタラプト処理のフローチャート、第3C図は第3B図の音振処理の詳細なフローチャート、第4図は時間にしたがったプログラムの流れを示す図、第5図は時間にしたがった処理の概要を示すタイムチャート、第6図は第2図のRAM34内に置かれる楽音生成用RAMのテーブルを示す図、第7図は第3C図の1つのチャンネル処理の詳細なフローチャート、第8図はエンベロープを示す図、第9図は波形ROMの波形データを示す図、第10図は時間にしたがった補間演算波形を示す図、第11図はインタラプト制御部の第1の構成例を示す図、第12図はインタラプト制御部の第2の構成例を示す図、第13図はインタラプト制御部のクロック入力のタイムチャートである。

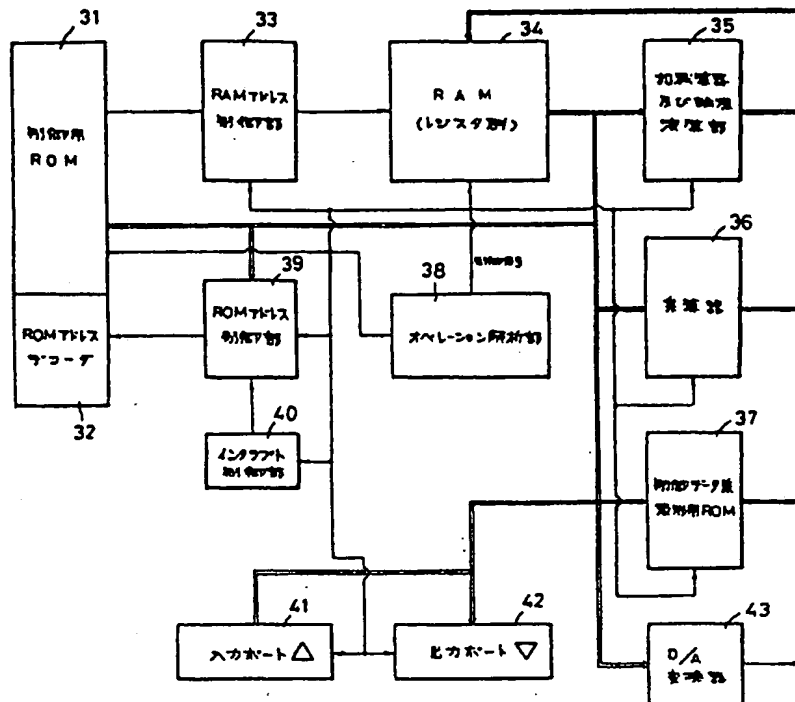
1……マイクロコンピュータ、31……制御用ROM、34……RAM、35……加減算器及び論理演算部、36……乗算器、37……制御データ波形ROM、38……オペレーション解析部、40……インタラプト制御部。

特許出願人 カシオ計算機株式会社

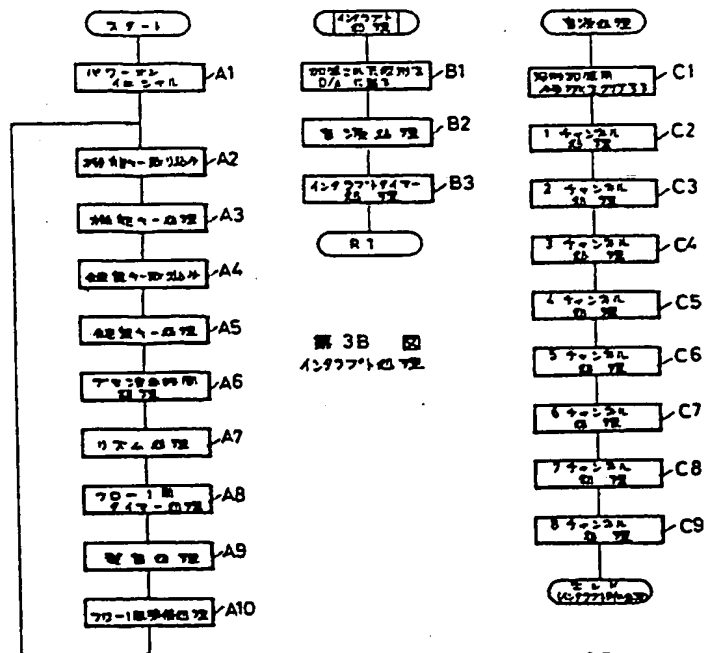
代理人 弁理士 町田 俊 正



第 1 図  
実施例の全体構成図



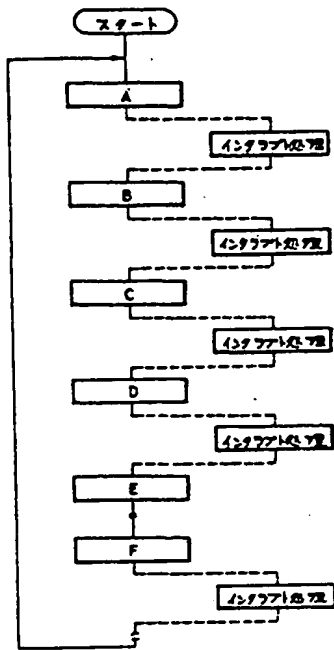
第 2 図  
本発明のマイクロコンピュータのブロック図



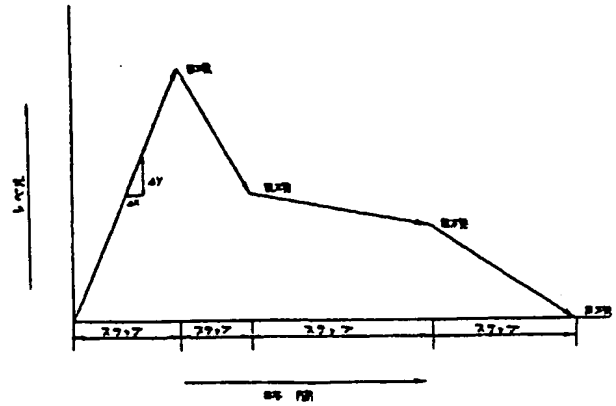
第 3A 図  
スタート

第 3B 図  
入力/出力

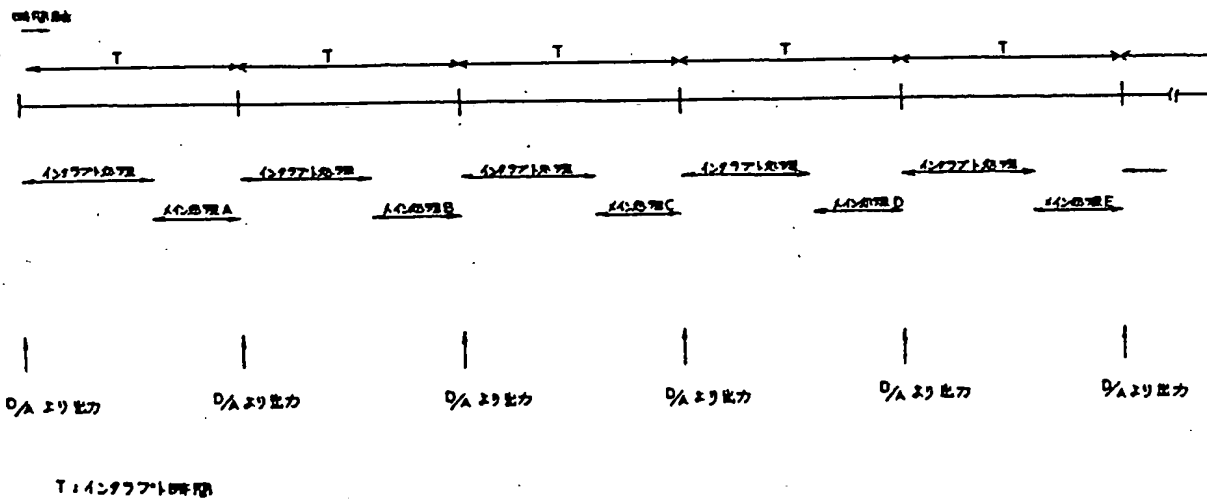
第 3C 図  
終了



第 4 図  
実行されるプログラムのフロー

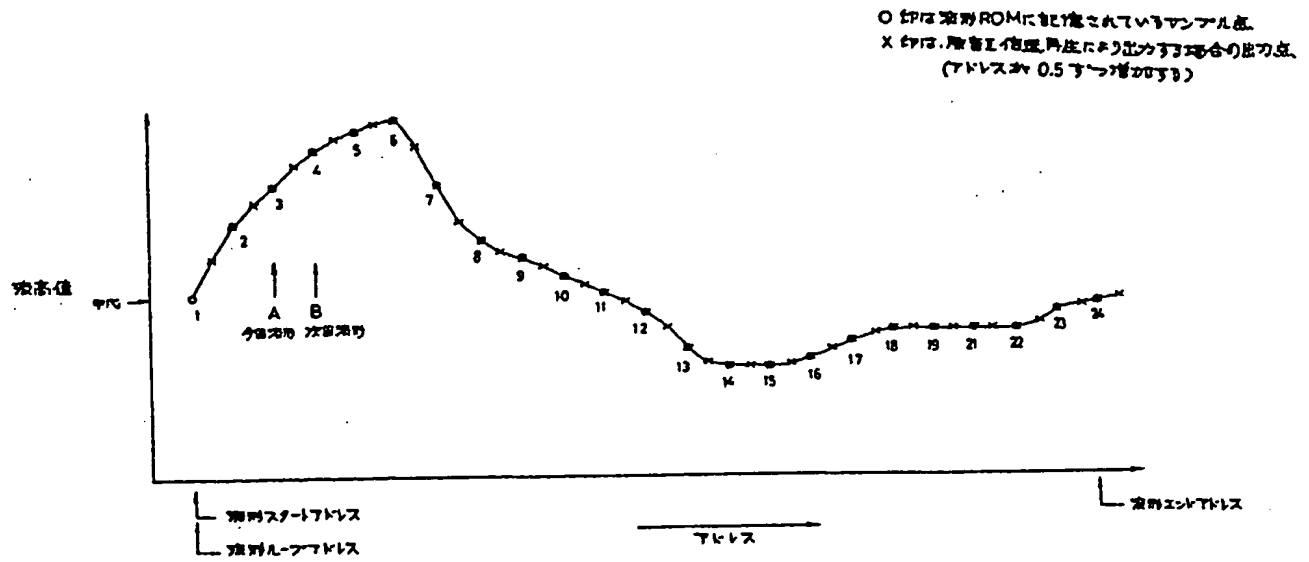


第 8 図  
エンベロープ

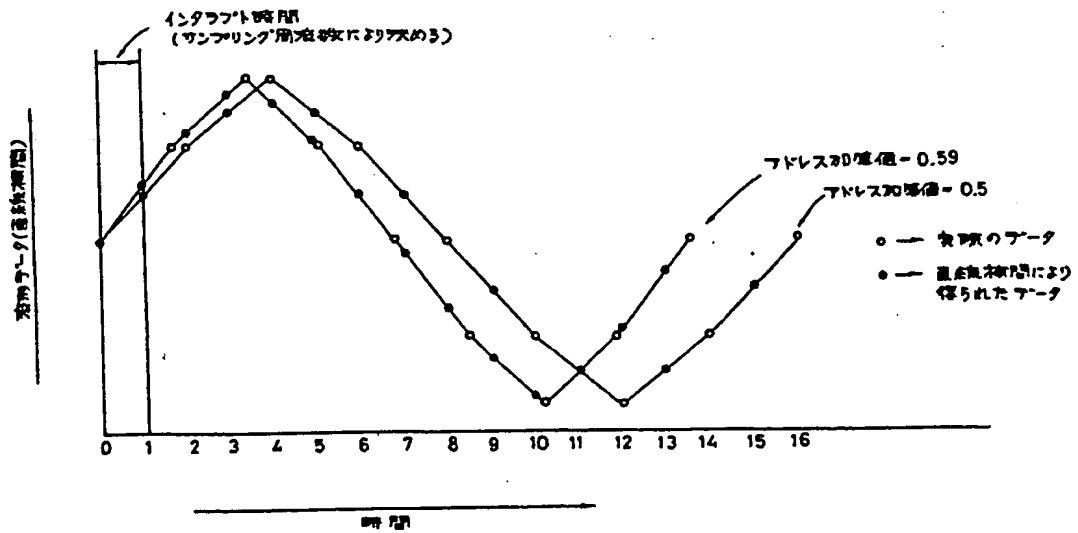


第 5 図  
動作のタイムチャート

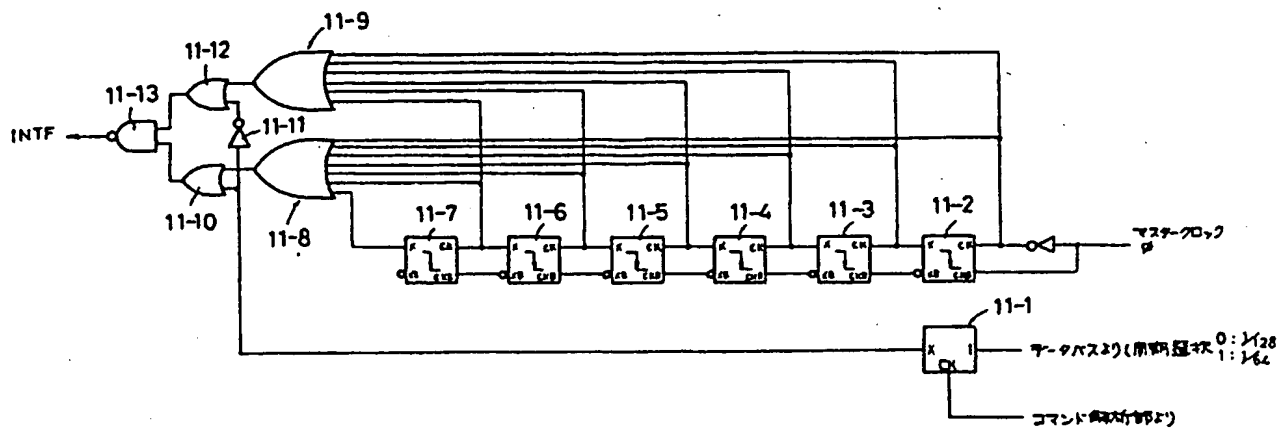




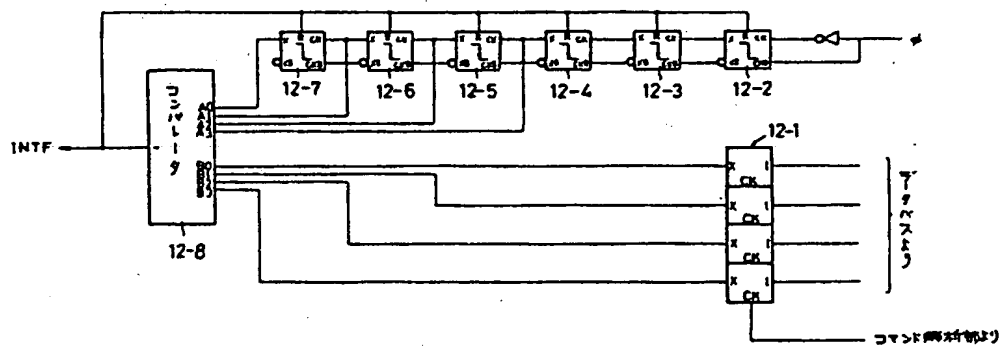
第 9 図  
波形データ



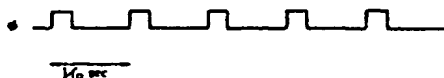
第 10 図  
置換時間による波形データ



第 11 図  
インタラプト処理部の構成例(1)



第 12 図  
インタラプト処理部の構成例(2)



第 13 図  
クロックのタイムチャート